

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-274405

(43)Date of publication of application : 26.09.2003

(51)Int.CI.

H04N 7/30  
G06F 13/362  
H04N 1/41

(21)Application number : 2002-068963

(71)Applicant : RICOH CO LTD

(22)Date of filing : 13.03.2002

(72)Inventor : KADOWAKI YUKIO

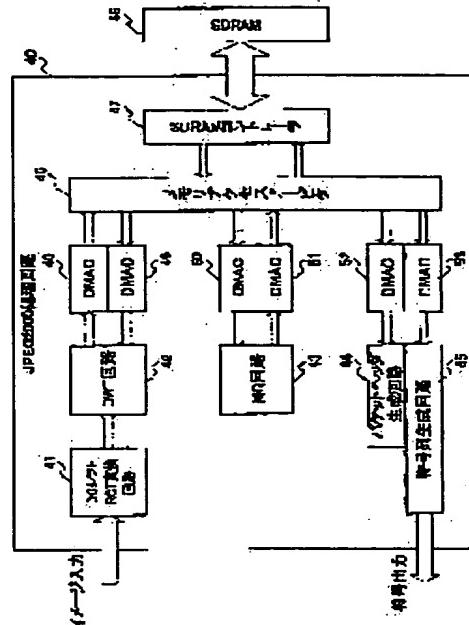
## (54) IMAGE PROCESSING APPARATUS AND IMAGE PROCESSING METHOD

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide an image processing apparatus for reducing the cost required for a memory in a processing with respect to encoding processing and/or decoding processing of an image and for attaining a high speed processing.

**SOLUTION:** This invention provides the image processing apparatus including: a wavelet transform circuit 42; an arithmetic encoding circuit 43; a packet header generating circuit 44; an encoding stream generating circuit 45; a memory for storing information read and/or written by the wavelet transform circuit 42; the arithmetic encoding circuit 43; the packet header generating circuit 44; and the encoding stream generating circuit 45; a memory access controller 47 for controlling the memory 55; and a memory access arbiter circuit for arbitrating a memory access request from the wavelet transform circuit 42; the arithmetic encoding circuit 43; the packet header generating circuit 44; and the encoding stream generating circuit 45 to the memory controller.

JP 2003-274405 A



### LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

[decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-274405

(P2003-274405A)

(43)公開日 平成15年9月26日(2003.9.26)

(51)Int.Cl.  
H 04 N 7/30  
G 06 F 13/362  
H 04 N 1/41

識別記号  
5 1 0

F I  
G 06 F 13/362  
H 04 N 1/41  
7/133

テマコード(参考)  
5 1 0 E 5 B 0 6 1  
B 5 C 0 5 9  
Z 5 C 0 7 8

審査請求 未請求 請求項の数 6 O L (全 18 頁)

(21)出願番号 特願2002-68963(P2002-68963)

(22)出願日 平成14年3月13日(2002.3.13)

(71)出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72)発明者 門脇 幸男

東京都大田区中馬込1丁目3番6号 株式会社リコー内

(74)代理人 100070150

弁理士 伊東 忠彦

最終頁に続く

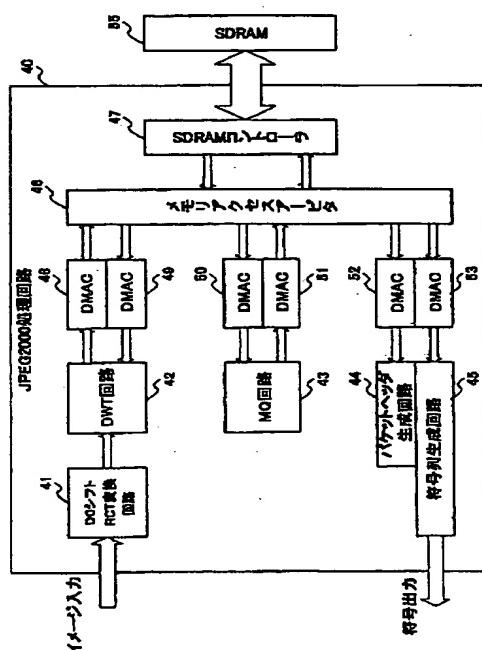
(54)【発明の名称】 画像処理装置及び画像処理方法

(57)【要約】

【課題】 画像の符号化処理及び/又は復号処理に関する処理におけるメモリに要するコストを削減し、更に、高速処理を可能とすることを目的とする。

【解決手段】 ウエーブレット変換回路42と、算術符号化回路43と、パケットヘッダ生成回路44と、符号列生成回路45と、ウエーブレット変換回路42、算術符号化回路43、パケットヘッダ生成回路44及び符号列生成回路45が書込み及び/又は読み出しする情報を格納するメモリ55と、該メモリ55の制御を行うメモリアクセスコントローラ47と、前記ウエーブレット変換回路42、前記算術符号化回路43、前記パケットヘッダ生成回路44、前記符号列生成回路45からの前記メモリコントローラへのメモリアクセス要求を調停するメモリアクセスアビタ回路とを有する画像処理装置。

JPEG2000の符号化処理回路を説明するための図



## 【特許請求の範囲】

【請求項1】 ウエーブレット変換回路と、算術符号化回路と、パケットの生成又は解析を行うパケット生成又は解析回路と、前記ウエーブレット変換回路、前記算術符号化回路、前記パケット生成又は解析回路が書き込み及び／又は読み出しする情報を格納するメモリと、該メモリの制御を行うメモリアクセスコントローラと、前記ウエーブレット変換回路、前記算術符号化回路、前記パケット生成又は解析回路からの前記メモリコントローラへのメモリアクセス要求を調停するメモリアクセスアービタ回路とを有する画像処理装置。

【請求項2】 前記メモリアクセスアービタ回路は、前記ウエーブレット変換回路のアクセスを優先させ、前記ウエーブレット変換回路からのアクセスがないときに、前記算術符号化回路、前記パケット生成又は解析回路からのアクセスを受け付けることを特徴とする請求項1記載の画像処理装置。

【請求項3】 前記ウエーブレット変換回路のアクセスのうち、レベル1の処理のアクセスを優先させてメモリアクセスさせることを特徴とする請求項1又は2記載の画像処理装置。

【請求項4】 コード化の単位であるコードブロック毎にコード化を行う場合、

コードブロックにおけるデータの有無を示すインクリジションデータを、内部レジスタに退避させることを特徴とする請求項1ないし3いずれか一項記載の画像処理装置。

【請求項5】 コード化の単位であるコードブロック毎にコード化を行う場合、

前記メモリ上に、前記コードブロック毎にポインタ領域を設け、

該ポインタ領域には、当該コードブロックのパケットデータが格納されているメモリのスタートアドレスと、当該コードブロックのデータ長と、当該コードブロックのゼロビットプレーン数と、コーディングバス数とを、前記コードブロック毎に格納し、

当該コードブロックのパケットデータは、前記メモリ上の前記ポインタに格納されているスタートアドレスから、格納されていることを特徴とする請求項1ないし4いずれか一項記載の画像処理装置。

【請求項6】 ウエーブレット変換回路と、算術符号化回路と、パケット生成又は解析回路と、メモリと、該メモリの制御を行うメモリアクセスコントローラと、前記ウエーブレット変換回路、前記算術符号化回路、前記パケット生成又は解析回路からの前記メモリコントローラへのメモリアクセス要求を調停するメモリアクセスアービタ回路とを有する画像処理装置における画像処理方法において、

前記メモリアクセスアービタ回路は、前記ウエーブレット変換回路のアクセスを優先させ、前記ウエーブレット

変換回路からのアクセスがないとき、前記算術符号化回路、前記パケット生成又は解析回路からのアクセスを受け付けることを特徴とする画像処理方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、画像処理装置及び画像処理方法に係り、特に、JPEG2000等に用いられる画像処理装置及び画像処理方法に関する。

## 【0002】

【従来の技術】 図1にJPEG2000の符号化の流れを示す。JPEG2000の符号化においては、画像データ1は、2次元離散ウエーブレット変換部(DWT)10、量子化器11、係数モデリング部12、算術符号器(算術符号化回路)13、符号形成部(パケットヘッダ生成回路及び符号列生成回路等を含む。)14の処理を経て、符号データ15が得られる。図2、図3にこれらの処理を模式的に描いた図を示す。なお、係数モデリング部12及び算術符号器13により、エントロピー符号化が行われる。

【0003】 まず、入力される画像に対して複数レベルの2次元離散ウエーブレット変換を行う。この時、処理する画像をタイルと呼ばれる複数の矩形ブロックに分ける場合が多い。画像を複数のタイルに分割する場合は、後に続く2次元離散ウエーブレット変換、量子化、エントロピー符号化はこのタイルの単位で処理する。図2は、画像データ1を $128 \times 128$ の大きさのタイルに分割し、レベル2の2次元離散ウエーブレット変換を行った場合の例を示す。 $128 \times 128$ の画像データ(1タイル)は、レベル2の2次元離散ウエーブレット変換10により、大きさが $32 \times 32$ の4つのサブバンド2LL、2HL、2LH、2HHと大きさが $64 \times 64$ の3つのサブバンド1HL、1LH、1HHのウエーブレット係数データ21に変換される。なお、1LL、1HLL、1LH、1HHはレベル1のサブバンドであり、2LL、2HL、2LH、2HHは、レベル2のサブバンドである。以下、3レベル以上のサブバンドについても同様に表現する。

【0004】 JPEG2000では、ウエーブレット変換のフィルタ係数としては、可逆、非可逆の2通りの係数が規定されている。可逆・非可逆を統一的に扱う場合は、可逆のフィルタ係数が用いられ、より高いレート又は良好なひずみ特性を実現する場合は非可逆のフィルタ係数を用いる。

【0005】 次に、ウエーブレット係数データに対して、次の式によりスカラーリングが行われる。

$$q = \text{sign}(a) \times \lfloor |a| / \Delta b \rfloor$$

ここで、 $\text{sign}(a)$ はウエーブレット係数データaの符号、 $|a|$ はaの絶対値、 $\Delta b$ はサブバンド毎に決められた量子化ステップ、 $\lfloor \cdot \rfloor$ はフロア関数を示す。ただし、可逆ウエーブレット係数を用いた場合は、この

スカラー量子化の処理は実施されない。

【0006】次に、ウエーブレット変換及び量子化した係数データをコードブロック毎に（サブバンドの大きさが、コードブロックよりも小さい場合は、各サブバンド毎に）エントロピー符号化していく。このエントロピー符号化は、後段の2値算術符号器13に与えるコンテキストを生成する係数モデリング部12での処理と、実際に符号化を行う算術符号器13での処理によりなされる。

【0007】図3は、各サブバンド毎にエントロピー符号化した場合におけるサブバンド2LLとその算術符号化器13の処理を示す。サブバンド2LLは、 $32 \times 32$ の画像サイズを有し、各点の画像（ウエーブレット変換及び量子化した係数データ）は、(N+3)ビットで量子化されている。上面が量子化ビットのMSBで、下面が量子化ビットのLSBである。図3では、上から二つのプレーンには、量子化ビットが存在しない。このプレーンをゼロビットプレーンという。

【0008】次に、算術符号器13での処理手順を示す。

(1) 一つのサブバンド（コードブロック毎にエントロピー符号化する場合は、コードブロック）の係数データを符号+絶対値に変換し、係数絶対値は、ビットプレーンに分割してMSB側からビットプレーン毎にエントロピー符号化を行う。図3(A)では、2LLのサブバンドの係数絶対値をビットプレーンに分割した様子を示す。

(2) 上位から数えて初めて有効なビット（0で無いビット）が出現するビットプレーンまでは、ゼロビットプレーンであるのでエントロピー符号化は行わず、初めて0で無いビットが出現したビットプレーンから、エントロピー符号化を開始する。図3(A)の例では、上位2プレーンがゼロビットプレーンであり、3プレーン目に初めて0で無いビットが出現し、このビットプレーンNからビットプレーン0までのN+1枚のビットプレーンについてエントロピー符号化を行う。

(3) エントロピー符号化を行うビットプレーンに関しては、基本的に1つのビットプレーンを3回スキャンして、符号化する。この符号化のためスキャンする過程をコーディングパスとして、各パスは以下のように呼ばれている。

【0009】① significance propagation pass

② magnitude refinement pass

③ cleanup pass

ビットプレーン上の各ビットは、特定の規則に従って分類され、そのビットの周辺ビットの状態から生成されたコンテキストを用いていずれかひとつのパスで符号化される。

【0010】図3(B)に示すように、最初に符号化する

ビットプレーンNは、cleanup passのみで符号化されるが、それ以降のビットプレーンは、それぞれ上に記した3つのパスで符号化される。すなわち、N+1個のビットプレーンが存在する場合、合計3N+1回のコーディングパスにより符号化が行われる。（4）次に、算術符号器13で、上記の各パスで発生する符号化ビットとそれに対応するコンテキストにより、エントロピー符号を生成する。こうして生成されたエントロピー符号（MQ符号）は、図1に示す符号形成部14における処理で、最終のJPEG2000のビットストリームとしてまとめられる。

【0011】符号形成部14では、まず、先の算術符号器13で生成された各パス単位で発生した符号を、複数パス毎にまとめる。このまとめられた単位をレイヤと呼ぶ。図4では、3N+1個のパス毎に算術符号器で生成された符号をレイヤ0からレイヤLまでのL+1個のレイヤにまとめている様子を示す。例えば、ビットプレーン2LLN、c、ビットプレーン2LLN-1、s、ビットプレーン2LLN-1、mをレイヤ0とし、…、ビットプレーン2LL0、s、ビットプレーン2LL0、m、プレーン2LL0、cをレイヤLとしている。なお、レイヤの選定は、これに限らず実施することができる。

【0012】さらに、図5に示すように、サブバンド、レイヤ毎にまとめられた符号を並べて最終のビットストリームが生成される。図5では、2LLのレイヤ0、2LLのレイヤ1、…、2LLのレイヤL、2HL、2LH、2HHのレイヤ0、…、2HL、2LH、2HHのレイヤL、1HL、1LH、1HHのレイヤ0、…、1HL、1LH、1HHのレイヤLのビットストリームとなる。ビットストリームは、重要なデータ（優先順位の高いデータ）を先に送信するように構成される。なお、レイヤは、任意に設定できる。例えば、すべてを一つのレイヤとしてもよい。

【0013】ここで、サブバンド、レイヤ毎のまとまりをパケットと呼び、各パケットは、それぞれのパケットの情報を示すパケットヘッダーと先ほどの算術符号部で生成されたエントロピー符号より成る。例えば、2HL、2LH、2HHのレイヤL34は、2HL、2LH、2HHのレイヤLのパケットヘッダー37と2HL、2LH、2HHのレイヤLのエントロピー符号38から構成される。したがって、ビットストリームは、1又は複数のパケットにより構成される。なお、各レベル毎に、LLのサブバンドで一つのパケットとし、HL、LH、HHのサブバンドで一つのパケットとしている。

【0014】ここでは、入力画像データが単色である場合を示したが、複数色（コンポーネント）を有する画像の場合、同様に各色（コンポーネント）毎にサブバンド毎、レイヤ毎のパケットを生成して規定された順番に並べることでJPEG2000のビットストリームを得る

ことができる。図6には、3色（コンポーネント0、1、2）より成る画像をJPEG2000で符号化した場合のビットストリームの例を示す。

【0015】JPEG2000の符号化方式は、一旦、符号化されたビットストリームを復号することなく符号状態のままで再度圧縮して、必要な圧縮率を得られるという特徴がある。これはこれまでに説明したようにJPEG2000の符号は、パケットと呼ばれるコンポーネント（色成分）、サブバンド（解像度）、レイヤ毎の符号を組み合わせて構成されているため、一旦、符号化を行った後で、その符号の圧縮率が所望の圧縮率より悪ければ、復号した際の画質面からみて優先順位の低いパケットの符号データを順次破棄していくことで、圧縮率を上げる操作が可能であることによる。

#### 【0016】

【発明が解決しようとする課題】従来のJPEG2000の符号化及び復号を行うJPEG2000装置では、エンコード時に算術符号化を行った後の符号データからパケットヘッダを作成する際、ひとつのタイル、又はプリシンクト（precincts：領域）、又はサブバンドのパケットデータをあらかじめ求めておき、このデータを用いてパケットヘッダを作成し、全体としての符号データを作成していた。この作業はハードで行うのは非常に煩雑になるため、多くの場合、ソフトウェア処理されていた。

【0017】ところで、ソフトで処理する場合、パケットデータは作成された順番にメモリに保持しておき、各パケットのデータ長、ゼロビットプレーン数、コーディングパス数を別途管理し、これらの情報を使用してパケットヘッダを作成していた。しかし、ソフト処理するためには、JPEG2000システムの中にプロセッサが必要となり、コストが上がる要因になっていた。また、ソフト処理なので、パケットデータの情報を管理する際に、ソフト上のポインタ（構造体）を使用することになり、必ずしもメモリを有効に使用しているとは限らず、ワーク用メモリを多く必要とし、これもコストアップの要因となっていた。

【0018】また、ハードウェアで処理を行う場合であっても、ウエーブレット係数と符号データは、別々のメモリを使用する方式が取られてきた。この場合、ハードウェアによる処理の並列化は簡単にできるが、扱う画像サイズが大きくなると、必要なメモリサイズも大きくなり、コストが大きくなるという問題がある。

【0019】本発明は、上記問題に鑑みなされたものであり、画像の符号化処理及び／又は復号処理に関する処理におけるメモリに要するコストを削減し、更に、高速処理を可能とすることを目的とするものである。

#### 【0020】

【課題を解決するための手段】上記課題を解決するために、本件発明は、以下の特徴を有する課題を解決するた

めの手段を採用している。

【0021】請求項1に記載された発明は、ウエーブレット変換回路と、算術符号化回路と、パケットの生成又は解析を行うパケット生成又は解析回路と、前記ウエーブレット変換回路、前記算術符号化回路、前記パケット生成又は解析回路が書き込み及び／又は読み出しうる情報を格納するメモリと、該メモリの制御を行うメモリアクセスコントローラと、前記ウエーブレット変換回路、前記算術符号化回路、前記パケット生成又は解析回路（例えば、パケットヘッダ生成回路、符号列生成回路、符号列解析回路等）からの前記メモリコントローラへのメモリアクセス要求を調停するメモリアクセスアービタ回路とを有する画像処理装置である。

【0022】請求項1に記載された発明によれば、ひとつのメモリにウエーブレット係数と符号データを格納することにより、大容量の安価な外部メモリを1個使用するだけでJPEG2000コーダ等の画像処理装置を実現することができる。

【0023】請求項2に記載された発明は、請求項1記載の画像処理装置において、前記メモリアクセスアービタ回路は、前記ウエーブレット変換回路のアクセスを優先させ、前記ウエーブレット変換回路からのアクセスがないときに、前記算術符号化回路、前記パケットヘッダ生成回路、又は、前記符号列生成回路からのアクセスを受け付けることを特徴とする。

【0024】請求項2に記載された発明によれば、メモリアクセスアービタでウエーブレット係数処理に関するアクセスを優先させることにより、処理速度が速まり、動画処理におけるリアルタイム処理が実現できる。

【0025】請求項3に記載された発明は、請求項1又は2記載の画像処理装置において、前記ウエーブレット変換回路のアクセスのうち、レベル1の処理のアクセスを優先させてメモリアクセスさせることを特徴とする。

【0026】請求項3に記載された発明によれば、メモリアクセスアービタでレベル1の処理のアクセスを優先させることにより、効率的な処理が可能となり、動画処理におけるリアルタイム処理が実現できる。

【0027】請求項4に記載された発明は、請求項1ないし3いずれか一項記載の画像処理装置において、コード化の単位であるコードブロック毎にコード化を行う場合、コードブロックにおけるデータの有無を示すインクルージョンデータを、内部レジスタに退避させることを特徴とする。

【0028】請求項4に記載された発明によれば、パケットヘッダ用ポインタを構成し、その中にパケットヘッダを構成するのに必要な情報をすべて備え、パケットデータのスタートアドレスで示されるメモリ上にパケットデータを配置する構成を取ることによって、ハードウェアのみでパケットヘッダ生成・解析を行う際に、パケットデータとそのパケットデータに付随する情報を効率よ

く管理することができ、使用するメモリ量を削減することができる。

【0029】請求項5に記載された発明は、請求項1ないし4いずれか一項記載の画像処理装置において、コード化の単位であるコードブロック毎にコード化を行う場合、前記メモリ上に、前記コードブロック毎にポインタ領域を設け、該ポインタ領域には、当該コードブロックのパケットデータが格納されているメモリのスタートアドレスと、当該コードブロックのデータ長と、当該コードブロックのゼロビットプレーン数と、コーディングパス数とを、前記コードブロック毎に格納し、当該コードブロックのパケットデータは、前記メモリ上の前記ポインタに格納されているスタートアドレスから、格納されていることを特徴とする。

【0030】請求項5に記載された発明によれば、コードブロックのインクルージョン情報のみメモリとは別に内部レジスタに退避させることにより、パケットヘッダの先頭に配置するタグツリーの生成を高速に容易に行うことができ、動画処理などで必要なリアルタイム性を確保することができる。

【0031】請求項6に記載された発明は、ウエーブレット変換回路と、算術符号化回路と、パケット生成又は解析回路と、メモリと、該メモリの制御を行うメモリアクセスコントローラと、前記ウエーブレット変換回路、前記算術符号化回路、前記パケット生成又は解析回路からの前記メモリコントローラへのメモリアクセス要求を調停するメモリアクセスアービタ回路とを有する画像処理装置における画像処理方法において、前記メモリアクセスアービタ回路は、前記ウエーブレット変換回路のアクセスを優先させ、前記ウエーブレット変換回路からのアクセスがないとき、前記算術符号化回路、前記パケット生成又は解析回路からのアクセスを受け付けることを特徴とする画像処理方法である。

【0032】請求項6に記載された発明は、請求項1～5記載の画像処理装置に適した画像処理方法を提供することができる。

### 【0033】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態を説明する。図7は、本実施の形態におけるJPEG2000の符号化処理回路40である。

(符号化処理回路) 図7の符号化処理回路40は、DCシフト・RCT(Reversible component Transformation)変換回路41、離散ウエーブレット変換回路(DWT回路)42、DMAC(Direct Memory Access Controller)48～53、算術符号化回路(MQ回路)であり、ここでは、係数モデリング部を含むものとしている。)43、パケットヘッダ生成回路44、符号列生成回路45、メモリアクセスアービタ46、SDRAM(Synchronous RAM)コントロー

ラ47、SDRAM55から構成されている。

【0034】DCシフト・RCT変換回路41におけるDCシフトは、符号無しで入力された画像データ(イメージ入力)のレベルをシフトして、0を中心に振幅変動する画像データに変換して、RCT変換回路、DWT回路42に入力するものである。DCシフト・RCT変換回路41におけるRCT変換は、RGB(赤・緑・青)データをYCbCr(輝度成分・色差成分)データに変換する。DWT回路42は、図1のDWT部10で説明した2次元離散ウエーブレット変換を行う。なお、この処理の詳細は後述する。DMAC48～53は、DWT回路42、MQ回路43、パケットヘッダ生成回路44、符号列生成回路45が、DMA転送を行うためのDMAコントローラである。MQ回路43は、算術符号処理を行う。パケットヘッダ生成回路44は、MQ回路43で生成されたパケットデータに対して、パケットヘッダを生成する。符号列生成回路45は、MQ回路43で生成された算術符号化後のパケットデータ及びパケットヘッダ生成回路44で生成されたパケットヘッダ等からビットストリームを生成して、出力する(符号出力)。

【0035】メモリアクセスアービタ46は、DWT回路42、MQ回路43、パケットヘッダ生成回路44、符号列生成回路45からのSDRAMコントローラへ47のメモリアクセス要求を調停する。SDRAMコントローラ47は、SDRAM55の書き込み及び読み出しを制御する。SDRAM55には、符号化処理回路装置で共有するひとつのメモリがあり、ウエーブレット係数、算術符号化後のパケットデータ、パケットデータ解析用のポインタ情報等を格納する。

【0036】図7の符号化処理回路40にイメージ入力信号が入力され、符号化されたビットストリームが、符号化処理回路40から出力される。

(復号処理回路) 図8は、本実施の形態におけるJPEG2000の復号処理回路60である。

【0037】図8の復号処理回路60は、DCシフト・RCT変換回路61、DWT回路62、DMAC65～69、MQ回路63、符号列解析回路69、メモリアクセスアービタ46、SDRAMコントローラ47、SDRAM55から構成されている。DCシフト・RCT変換回路61は、図1の符号化処理回路のDCシフト・RCT変換回路41の逆の処理を行う。つまり、シフトしたレベルを元に戻す。DWT回路62は、図1の符号化処理回路のDWT回路41の逆の処理を行う。つまり、2次元離散ウエーブレット逆変換を行う。DMAC65～69は、図1の符号化処理回路のDMAC48～53と同じく、DWT回路62、MQ回路63、符号列解析回路64が、DMA転送を行うためのDMAコントローラである。MQ回路63は、図1の符号化処理回路のMQ回路43の逆の処理を行う。つまり、算術復号を行う。符号列解析回路69は、受信したビットストリーム

の符号列を解析して、パケットデータ及びパケットヘッダを出力する。

【0038】メモリアクセスアービタ46、SDRAMコントローラ47、SDRAM55は、図1の符号化処理回路のもと同じである。

【0039】なお、DCシフト・RCT変換回路61、DWT回路62、DMAC65～68、MQ回路63は、それぞれ、図1の符号化処理回路のDCシフト・RCT変換回路41、DWT回路42、DMAC48～51、MQ回路43と一緒に構成してもよい。

【0040】図8の符号化処理回路60にビットストリームである符号が入力され、符号化処理回路60から復号されたイメージ信号が出力される。

(DWT変換：エンコード) 図7において、DCシフト・RCT変換回路41からの出力に対して、DWT回路42は、2次元離散ウェーブレット変換を行う。

【0041】ウェーブレット変換を行う場合、最初に場面全体に対して、垂直成分と水平成分に対してそれぞれハイパスフィルタとローパスフィルタをかけて1/2のデータにした結果を4個出力する。この結果を並べた結果を図9に示す。ここで、1LHはウェーブレット変換レベル1の垂直方向ローパス、水平方向ローパス成分を示し、1HLはウェーブレット変換レベル1の垂直方向ハイパス、水平方向ローパス成分を示し、1LHはウェーブレット変換レベル1の垂直方向ローパス、水平方向ハイパス成分を示し、1HHは垂直方向ハイパス、水平方向ハイパス成分を示す。

【0042】JPEG2000で規定されているウェーブレット変換は、R5:3フィルタとIR9:7フィルタがある。ここでは、R5:3フィルタの説明を行う。R5:3フィルタの処理を図15の構成例に基づいて説明する。

$$Y(1) = X(1) - [ (X(0) + X(2)) / 2 ] \quad \dots \quad (1)$$

の処理を行う。このデータY(1)は、ラインバッファ84、フリップフロップ回路87及びローパス処理回路79に順次供給される。

【0048】図16(C)に示されているように、このデータY(1)が、入力されたタイルデータを垂直方向

$$Y(3) = X(3) - [ (X(2) + X(4)) / 2 ] \quad \dots \quad (2)$$

の処理を行う。このデータY(3)は、データY(1)と同様に、ラインバッファ84、フリップフロップ回路87及びローパス処理回路79に順次供給される。

【0050】図16(E)に示されているように、このデータY(1)が、入力されたタイルデータを垂直方向で周波数領域に変換したときの、第2番目のハイパス成

$$Y(0) = X(0) - [ (Y(-1) + Y(1) + 2) / 4 ] \quad \dots \quad (3)$$

の処理を行う。このデータY(0)は、フリップフロッ

【0043】図15に示すウェーブレット変換のエンコード装置は、ラインバッファ82J～84、ハイパス処理を行うハイパス処理回路76～78、ローパス処理を行うローパス処理回路79～81、クロック端子を有するフリップフロップ回路85～90、入力端子71及び出力端子72～75から構成されている。

【0044】ラインバッファ82には、図16(A)に示されているように、入力されたタイルの最初の1ライン分のデータX(0)が順次蓄積される。ラインバッファ83には、図16(B)に示されているように、タイルの2ライン目の1ライン分のデータX(1)が順次蓄積される。また、入力されたタイルの3番目の1ライン分のデータX(3)は、ラインバッファ82に蓄積され、入力されたタイルの4番目の1ライン分のデータX(4)は、ラインバッファ83に蓄積される。なお、入力されたタイルの3番目の1ライン分のデータX(3)が、ラインバッファ82に蓄積されるとき、ラインバッファ82からは、蓄積の直前に、最初の1ライン分のデータX(0)が読み出されて、ローパス処理回路79に供給される。以下同様のタイミングで、データX(2)、X(4)、...が順次読み出されて、ローパス処理回路79に供給される。

【0045】以下、入力されたタイルの奇数番目の1ライン分のデータは、ラインバッファ82に蓄積され、入力されたタイルの偶数番目の1ライン分のデータは、ラインバッファ83に蓄積される。

【0046】次に、ハイパス処理回路76及びローパス処理79によるレベル1の垂直処理について説明する。

【0047】ハイパス処理回路76は、データX(0)、データX(1)及びデータX(2)に基づいて、

で周波数領域に変換したときの、第1番目のハイパス成分である。

【0049】同様にして、ハイパス処理回路76は、データX(2)、データX(3)及びデータX(4)に基づいて、

$$Y(4) = X(4) - [ (X(2) + X(3)) / 2 ] \quad \dots \quad (4)$$

分である。

【0051】以下順に、Y(5)、Y(7)、...、Y(127)を作成する。

【0052】ローパス処理回路79は、データX(0)、データY(-1)及びデータY(1)に基づいて、

$$Y(0) = X(0) - [ (Y(-1) + Y(1) + 2) / 4 ] \quad \dots \quad (5)$$

フリップフロップ回路85及びハイパス処理回路77に順次供給され

る。

【0053】図16(C)に示されているように、このデータY(0)が、入力されたタイルデータを周波数領域に変換したときの、第1番目のローパス成分である。

$$Y(2) = X(2) - [ (Y(1) + Y(3) + 2) / 4 ] \quad \dots \quad (4)$$

の処理を行う。このデータY(2)は、フリップフロップ回路85及びハイパス処理回路77に順次供給される。

【0055】このデータY(2)は、データY(0)と同様に、フリップフロップ回路85及びハイパス処理回路77に順次供給される。

【0056】図16(E)に示されているように、このデータY(1)が、入力されたタイルデータを垂直方向で周波数領域に変換したときの、第2番目のハイパス成分である。

【0057】以下順に、Y(4)、Y(6)、...、Y(126)を作成する。

【0058】次に、ハイパス処理回路77、78及びローパス処理80、81によるレベル1の垂直処理に連続して行なわれるレベル1の水平処理について説明する。

【0059】ローパス処理回路79により、図17

$$Z(1) = Y(0, 1) - [ (Y(0, 0) + Y(0, 2)) / 2 ] \quad \dots \quad (5)$$

の処理を行う。このデータZ(1)は、フリップフロップ回路89及びローパス処理回路80に供給される。

【0061】図17(C)に示されているように、このデータZ(1)が、1ライン目の垂直方向の低周波数成

$$Z(3) = Y(0, 3) - [ (Y(0, 2) + Y(0, 4)) / 2 ] \quad \dots \quad (6)$$

の処理を行う。このデータZ(3)は、フリップフロップ回路89及びローパス処理回路80に供給される。

【0063】図17(E)に示されているように、このデータZ(3)が、1ライン目の垂直方向の低周波数成分の、水平成分の第2番目のハイパス成分である。

【0064】以下順に、Z(5)、Z(7)、...、Z(127)を作成する。

【0065】また、同様に、2ライン目の以下の、垂直方向の低周波数成分の、水平成分のハイパス成分を作成する。その結果、端子73から1LHデータが得られる。

$$Z(0) = Y(0, 0) - [ (Z(-1) + Z(1) + 2) / 4 ] \quad \dots \quad (7)$$

の処理を行う。なお、データZ(-1)は、ミラー成分としてデータZ(1)が使用される。

【0069】図17(C)に示されているように、このデータZ(0)が、1ライン目の垂直方向の低周波数成

$$Z(2) = Y(0, 2) - [ (Z(1) + Z(3) + 2) / 4 ] \quad \dots \quad (8)$$

の処理を行う。なお、データZ(-1)は、ミラー成分と

【0054】同様にして、ローパス処理回路79は、データX(2)、データY(1)及びデータY(3)に基づいて、

$$Y(2) = X(2) - [ (Y(1) + Y(3) + 2) / 4 ] \quad \dots \quad (4)$$

(A)に示す1ライン目における第1番目の垂直方向の低周波数成分Y(0, 0)が生成され、フリップフロップ回路85に供給される。同様に、ローパス処理回路79により、図17(B)に示す1ライン目の垂直方向の第2番目の低周波数成分Y(0, 1)、1ライン目の垂直方向の第3番目の低周波数成分Y(0, 2)、...が生成され、フリップフロップ回路85に供給される。なお、フリップフロップ回路85は、データが入力されるタイミングで、それまでのデータをフリップフロップ回路86に供給する。同様に、フリップフロップ回路86~90は、データが入力されるタイミングで、それまでのデータを出力する。

【0060】ハイパス処理回路77は、データY(0, 0)、データY(0, 1)、及びデータY(0, 2)に基づいて、

分の、水平成分の第1番目のハイパス成分である。

【0062】同様にして、ハイパス処理回路77は、データY(0, 2)、データY(0, 3)、及びデータY(0, 4)に基づいて、

【0066】同様に、垂直方向の高周波数成分であるデータY(1)、データY(3)、...、データY(127)について、フリップフロップ回路87、88及びハイパス処理回路78が、ハイパス処理を行って、端子75から1HHデータが得られる。

【0067】ハイパス処理回路77により、図17(A)に示す1ライン目の垂直方向の高周波数成分Y(1, 0)が生成され、フリップフロップ回路89及びローパス処理回路80に供給される。

【0068】ローパス処理回路80は、データY(0, 0)、データZ(-1)及びデータZ(1)に基づいて、

分の、水平成分の第1番目のローパス成分である。

【0070】同様にして、ローパス処理回路80は、データY(0, 2)、データZ(1)及びデータZ(3)に基づいて、

してデータZ(1)が使用される。

【0071】図17(E)に示されているように、このデータZ(2)が、1ライン目の垂直方向の低周波数成分の、水平成分の第2番目のローパス成分である。

【0072】以下順に、Z(4)、Z(6)、...、Z(126)を作成する。

【0073】また、同様に、2ライン目の以下の、垂直方向の低周波数成分の、水平成分のローパス成分を作成する。その結果、端子72から1LHデータが得られる。

【0074】同様に、フリップフロップ回路90及びローパス処理回路81により、1HLデータが作成され、端子75から1HLデータを得ることができる。

【0075】なお、図15の回路は、データの流れに即して、イネーブル信号を伝播させるものであり、データフロー型パイプライン制御が行なわれている。

【0076】また、図15の回路は、データフロー内の有効期間は全てマスタークロックで2クロックである。この2クロックの内、演算処理は必ず後半クロックで行う。

【0077】また、ラインバッファ処理するときは、前サイクルでパイプラインリードを開始し、後半クロックでライトを行う。ラインバッファは、2ポートRAMを使用するのではなく、時分割処理するので、一つのデータ処理には必ず2クロック必要となる。また、ラインバッファ82、84は、データを出力しながら新しいデータを入力している。

【0078】また、ワークメモリへのライトを2クロックで1データ処理するので、全てのフィルタに入力されるタイミングは、回路全体で管理する必要がある。

【0079】このような構成により、入力端子71に印加されたタイル入力データについて、最初に水平方向のハイパスフィルタ処理を行い、その結果を用いて水平方向のローパスフィルタの処理を行う。同様にして、水平方向のハイパスフィルタの処理結果を用いて、垂直方向のハイパスフィルタおよび垂直方向のローパスフィルタ処理をそれぞれ行い、また、水平方向のローパスフィルタの処理結果を用いて垂直方向のハイパスフィルタおよび垂直方向のローパスフィルタ処理をそれぞれ行う。その結果、端子72、73、74、75から、図9に示されているように、ウェーブレット変換された1LH、1LH、1HL、1HHの4種類の成分に分割されたウェーブレット係数が得られる。

【0080】この1LHのウェーブレット係数に対して、1レベルの処理と同じようにウェーブレット変換を行うと図10に示すように2レベルのウェーブレット係数を得ることができる。同様に、図10の2LHに対してウェーブレット変換を行うことで図11に示すように3レベルのウェーブレット係数を得ることができる。同様にして、図11の3LHのウェーブレット係数に対してウェーブレット変換を行うことで、図12に示すよう

な4レベルのウェーブレット係数を得ることができる。同様にして、図12の4LHのウェーブレット係数に対してウェーブレット変換を行うことで、図13に示す5レベルのウェーブレット係数を得ることができる。図14に、最終的なウェーブレット変換のエンコードの結果を示す。

【0081】図18により、図15に示すウェーブレット変換のエンコード装置におけるハイパス処理回路の例を説明する。

【0082】図18のハイパス処理回路は、ハイパス処理及びその逆変換を行う。図18のハイパス処理回路は、マルチプレクサ100～104、加算器105、106、反転回路107、クロック端子を有するフリップフロップ回路108、端子110～123を有している。

【0083】式(1)のハイパス処理演算を行う場合は、端子114にデータX(0)が供給され、端子115にデータX(2)が供給され、端子122にデータX(1)が供給される。このとき、マルチプレクサ100～103は、端子112に印加されたコード・デコード切替え信号に基づいて、端子114、端子115、端子122からのデータを選択する。また、マルチプレクサ103は、端子112に印加されたコード・デコード切替え信号に基づいて、反転回路107からの信号を選択する。

【0084】なお、端子110、111には、境界処理用のミラーデータが供給される。また、端子113には、演算イネーブル信号が印加され、加算器106の信号はこの演算イネーブル信号が印加された時点の信号にラッチされる。

【0085】図19に、図15に示すウェーブレット変換のエンコード装置におけるローパス処理回路の例を示す。

【0086】図19のローパス処理回路は、ローパス処理及びその逆変換を行う。図19のローパス処理回路は、マルチプレクサ200～204、加算器205、206、224、反転回路207、クロック端子を有するフリップフロップ回路208、端子210～223を有している。

【0087】式(1)のローパス処理演算を行う場合は、端子214にデータY(-1)が供給され、端子215にデータY(1)が供給され、端子222にデータX(0)が供給される。このとき、マルチプレクサ200～203は、端子212に印加されたコード・デコード切替え信号に基づいて、端子214、端子215、端子222からのデータを選択する。また、マルチプレクサ203は、端子212に印加されたコード・デコード切替え信号に基づいて、反転回路207からの信号を選択する。

【0088】なお、端子210、211には、境界処理

用のミラーデータが供給される。また、端子213には、演算イネーブル信号が印加され、加算器206の信号はこの演算イネーブル信号が印加された時点の信号にラッピングされる。

【0089】以上のようにして複数レベルのウエーブレット係数を求めるが、このとき、すべてのウエーブレット係数を一度バッファに格納しておく必要がある。また、特に1レベルのウエーブレット係数を求めるときに、ウエーブレット係数を対比するメモリ処理が忙しいために、入力イメージにウエイトが必要になる場合があると、入力データによってはウエイトをかけてしまうとデータが失われてしまう場合がある。このため、少なくとも1レベルのウエーブレット変換処理を行う場合は、特に動画処理の場合は、入力画像がくるスピード以内で処理を完了させる必要がある。このとき、図7に示すように、ひとつのメモリでウエーブレット係数と符号メモリを共有させる場合、ウエーブレット係数処理のアクセスの優先順位を高める必要がある。そして特に、1レベルのウエーブレット処理のメモリアクセスの優先順位を高める必要がある。

(DWT変換：デコード) ウエーブレット変換のエンコード装置を説明するための図である図15に対応するウエーブレット変換のデコード装置の例を図20に示す。

【0090】また、図21に、ウエーブレット変換のデコード垂直方向処理を説明するための図を示し、図22に、ウエーブレット変換のデコード水平方向処理を説明するための図を示す。図21及び図22は、ウエーブレット変換のエンコード垂直方向処理を説明するための図16と、ウエーブレット変換のエンコード水平方向処理を説明するための図17に対応する。また、図20～図22は、図15～図17のウエーブレット変換のエンコード装置の例に準じて説明できるので、これらの説明は省略する。

(パケットデータの格納) また、JPEG2000における算術符号化の詳細な説明はここでは行わない。算術符号化は、ウエーブレット係数上のL<sub>L</sub>、H<sub>L</sub>、L<sub>H</sub>、H<sub>H</sub>のサブバンド単位で行うが、処理範囲が大きくなるために、JPEG2000ではコードブロックという単位で処理を行うようになっている。コードブロックサイズがたとえば $64 \times 64$ と決めると、サブバンドのサイズが $64 \times 64$ より小さいときは、サブバンドのサイズで処理を行い、サブバンドのサイズがコードブロックより大きい場合は、サブバンドをコードブロックの大きさで分割し、コードブロック単位で符号化を行う。イメージサイズが $2048 \times 2048$ で5レベルまでウエーブレット変換を行い、コードブロックサイズが $64 \times 64$ の場合の例を図23に示す。この場合、1レベルから5レベルまでのサブバンドのサイズは1024、512、128、64、32となり、1から3レベルまではサブバンドサイズがコードブロックより大きいので、これら

のサブバンドはコードブロックサイズで分割され、4レベルと5レベルに関してはサブバンドの大きさがコードブロックサイズと同じか小さいのでサブバンドの分割は行われていない。算術符号化はこのようにコードブロックサイズとサブバンドのどちらか小さいほうのサイズで行われる。

【0091】また、上述したように、算術符号化を行うことで、符号データが生成される。これを符号列生成することでJPEG2000の符号データが作成される。JPEG2000ではコードブロックごとの符号データをパケットと呼び、ひとつのパケットに対して、パケットヘッダを付加するようになっている。パケットの単位は、L<sub>L</sub>サブバンドで1パケットを構成し、それ以外はH<sub>L</sub>、L<sub>H</sub>、H<sub>H</sub>の3つのサブバンドをまとめて一つのパケットとして扱う。

【0092】本実施の形態では、パケットデータの長さとスタートアドレスを示したポインタを使用してデータの管理を行う。パケットの単位はサブバンド、コードブロック、プリシンクト、レイヤで決まる。ひとつのパケットには1レイヤのL<sub>L</sub>又はH<sub>L</sub>、L<sub>H</sub>、H<sub>H</sub>のサブバンド成分が含まれる。ただし、プリシンクトサイズがサブバンドより小さい場合、パケットはプリシンクトのサイズで区切られる。パケットの単位は上記の方法で決まるが、各コードブロックの各レイヤの情報はパケットヘッダ生成に使用される。MQ符号化を行った後、コードブロック、レイヤ、コンポーネント、レベルの符号データに対して、識別子を付け、スタートアドレス、データ長、ゼロビットプレーン数、コーディングバス数をポインタに格納した後、設定されたスタートアドレスから符号データを格納する。パケットデータ用ポインタの例を図24に示す。図24の例では、アドレスが32ビット(4バイト)、データ長が16ビット(2バイト)、ゼロビットプレーン数とコーディングバス数がそれぞれ8ビット(1バイト)で構成されている。また、図24のパケットポインタとパケットデータのメモリマッピングの例を、図25に示す。

【0093】各コードブロック、各レイヤのポインタがどこになるかはポインタアドレスで管理されている。各コードブロック、各レイヤのポインタはMQコーダの処理順に配列する。MQコーダの処理順(0～258)の例を図23に示す。図23の例ではあるひとつのコンポーネントの、 $2048 \times 2048$ のフレームデータに対して、5レベルのウエーブレット変換を行い、コードブロックサイズが $64 \times 64$ の場合を示している。各コードブロック、各レイヤにデータが無い場合にもポインタは配置する。

【0094】ウエーブレット変換のエンコード装置は、1フレームの符号処理が完了すると、パケットヘッダの生成を開始する。各コードブロックの各レイヤのポインタ内にパケットヘッダを生成するのに必要な情報が含まれる。

れている。パケットヘッダを作るときは、パケットデータのインクルージョンを最初に決める。パケットヘッダのインクルージョンを求める場合、JPEG2000ではタグツリーという概念をとっている。タグツリーを使用すると、コードブロック内にデータが含まれないとときにパケットヘッダに使用するビット数を省略できる場合がある。しかし、タグツリーを作る作業は煩雑で、パケット内のすべてのコードブロックの情報が揃わないとツリーが完成しない。

【0095】次に、パケットヘッダのインクルージョンを決める方法を説明する。

(1) 対象となるサブバンド内のコードブロックの配置を確認する。ひとつのパケットに含まれるコードブロックの数はプリシンクトサイズによって決定される。一般的には、図23のように、プリシンクトサイズは一番大きいサブバンドより大きく設定される。しかし、図26のように、サブバンドの大きさよりプリシンクトサイズが小さい場合、パケットはプリシンクト内のコードブロックで構成される。

(2) ひとつのサブバンド内に1つ以上のコードブロックが含まれる場合、図27に示すように、タグツリーを使用する。タグツリーは最大4個のコードブロック又はタグから構成されている。サブバンド内のデータが多くなるとタグツリーの段数も増えていく。

(3) ひとつのサブバンド、又はプリシンクトにおけるトップのタグにデータがない場合、そのサブバンド、又はプリシンクトにデータがひとつも含まれないことになる。パケットはLL成分以外はHL、LH、HH成分で構成されている。LL成分においてトップのタグにデータがない場合はパケットにデータがないことを示す(No packet)。その他のサブバンドの場合、HL、LH、HHすべての成分のトップのタグにデータがない場合にパケットにデータがないことになる。それ以外の場合は、パケットに有効なデータが含まれることを示す。

(4) パケットのインクルージョンは、最初にパケットインクルージョンを示すために1ビットを使用する。このビットに1が立っているときはパケットにデータがあることを示す。このビットが0の時はパケット内にデータがない状態を示す。

(5) 次に、パケット内のコードブロックのインクルージョンをラスタ順に調べていく。このとき、LL成分以外のパケットはHL、LH、HHのサブバンドの順番でコードブロックのインクルージョンを調べていく。

(6) コードブロックのインクルージョンを調べる方法は、最初のコードブロックに関連するタグをトップから順番に並べていく。図27を例にとって説明する。図27はHL成分のサブバンドを表している。最初に、パケット内のデータの有無を調べるビットの後に、コードブロック67のインクルージョンを調べる。コードブロッ

ク67のインクルージョンを調べる前に、コードブロック67はタグ1000に属しているので、タグ1000を先に調べる。しかし、タグ1000はタグ1100に属しているので、タグ1100を先に調べる。しかし、タグ1100はタグ1200に属しているのでタグ1200を先に調べる。このようにして、コードブロック67のインクルージョンを調べるときはコードブロック67に関連するタグをトップから順に調べることになる。このとき、トップのタグ1200が0の時、サブバンドHLのすべてのコードブロックにデータが無いことを示しているのでコードブロック67から130までデータがないことを示すことになる。この場合、パケットは次にサブバンドLHのコードブロックのインクルージョンを調べることになる。もし、タグ1200が1でサブバンドにデータがあることが示されている場合、次にタグ1100を調べる。タグ1100が0の場合、このタグに関連するコードブロックすべてにデータがないことを示すので、次にコードブロック71のインクルージョンを調べることになる。タグ1100が1でデータがあることを示していると、次にタグ1000を調べる。タグ1000が0でデータがないことを示していると、関連するコードブロックにデータがないことを示すので、次にコードブロック69のインクルージョンを調べることになる。タグ1000が1でデータがある場合、初めてコードブロック67のインクルージョンを調べることになる。このようにして、各コードブロックのインクルージョンを調べていく。

(7) コードブロック67のインクルージョンが1でパケットヘッダを作成した後、コードブロック68のインクルージョンを調べる。このとき、コードブロック68に関連するタグ1200、1100、1000は、すでに調べられているので、このときはコードブロック68のインクルージョンを直接調べる。次に、コードブロック69のインクルージョンを調べる場合、タグ1200、1100は、既に調べられているが関連するタグ1001がまだ調べられていないので、タグ1001を調べてからコードブロック69のインクルージョンを調べることになる。

(8) 図28にタグツリーを使用したときのインクルージョン構成例を示す。最初タグ1200を見る。この場合はタグにデータがあるので、このサブバンドにデータがあることを示している。(図28(A))。次に、ひとつ下の階層のタグ1100、1101、1102、1103を調べる(図28(B))。タグ1100、1103にはデータがあるが1101、1102にはデータがないとする。この場合、タグにデータがない部分に関連するコードブロックにはすべてデータがないことがわかる。次に、データがある部分のタグの下を調べる。この場合、タグ1100の下には1000、1001、1002、1003があるがこのうち1000、1001、100

2にデータがあるが1003にデータがないとする。また、タグ1103の下にも1012、1013、1014、1015のタグがあるがこのうち1013、1014にデータがあるが1012、1015のタグにはデータがないとする。次にデータがあるタグの下を調べる(図28(D))。この場合、タグの下にはコードブロックになるので、各コードブロックのインクルージョンを調べることになる。タグツリーを使用することで、データがないコードブロックひとつに対して、コードブロックの有無を示すインクルージョンビットを1ビット節約できる場合が発生する。

【0096】以上のように、タグツリーを構成するためには、サブバンドのすべてのコードブロックのインクルージョンの情報が必要になる。しかし、ひとつのメモリ内に、パケットポインタという形でインクルージョンの情報を格納していると、タグツリーを作るときに、すべてのコードブロックのインクルージョン情報をメモリリードしなければならず、メモリアクセスに時間がかかる。また、タグツリーのインクルージョンはパケットヘッダの最初にくるので、インクルージョンが決まらなければそのあとのパケットヘッダ生成を開始できない問題がある。インクルージョンが決まると、その後のパケットデータは、コードブロックごとの情報をメモリリードしていけばよいので効率よくパケットヘッダ作成ができる。この問題を回避するために、算術符号化が完了した後、各コードブロックのデータと情報をメモリに書き込む際、コードブロックのインクルージョンに関する情報のみ、メモリではなく、内部レジスタに格納し、すべてのコードブロックの算術符号化が終了したときに、すべてのコードブロックのインクルージョンの情報を使用して簡単にタグツリーを構成できるようにすることで、パケット生成の時間を短縮できる。コードブロックのインクルージョン情報はコードブロックひとつについて1ビットですむのでハード量もほとんど増加しない。

【0097】JPEG2000の動画のエンコード時においては、入力されるイメージデータは一定の時間間隔で入力されるため、各入力画像を所定の期間内に処理しなければならないが、本発明の実施の形態によれば、迅速な処理が可能であり、リアルタイム処理が可能となる。

【0098】

【発明の効果】上述の如く本発明によれば、画像の符号化処理及び／又は復号処理に関する処理におけるメモリに要するコストを削減し、更に、高速処理を行うことができる。

【0099】

【図面の簡単な説明】

【図1】JPEG2000の符号化の流れを説明するための図である。

【図2】画像データ1を128×128の大きさのタイルに分割し、レベル2の2次元離散ウエーブレット変換

を行った場合の例を説明するための図である。

【図3】各サブバンド毎にエントロピー符号化した場合におけるサブバンド $2Ll$ とその算術符号化器の処理を説明するための図である。

【図4】 $3N+1$ 個のバス毎に算術符号化器で生成された符号をレイヤ0からレイヤ $l+1$ までの $l+1$ 個のレイヤにまとめている様子を示す図である。

【図5】ビットストリームを説明するための図である。

【図6】3色(コンポーネント0、1、2)より成る画像をJPEG2000で符号化した場合のビットストリームの図である。

【図7】JPEG2000の符号化処理回路を説明するための図である。

【図8】JPEG2000の復号処理回路を説明するための図である。

【図9】垂直成分と水平成分に対してそれぞれハイパスフィルタとローパスフィルタをかけて $1/2$ のデータにした結果を説明するための図である。

【図10】2レベルのウエーブレット係数を説明するための図である。

【図11】3レベルのウエーブレット係数を説明するための図である。

【図12】4レベルのウエーブレット係数を説明するための図である。

【図13】5レベルのウエーブレット係数を説明するための図である。

【図14】最終的なウエーブレット変換のエンコードの結果を示す図である。

【図15】ウエーブレット変換のエンコード装置を説明するための図である。

【図16】ウエーブレット変換のエンコード垂直方向処理を説明するための図である。

【図17】ウエーブレット変換のエンコード水平方向処理を説明するための図である。

【図18】ハイパス処理回路の例を説明するための図である。

【図19】ローパス処理回路の例を説明するための図である。

【図20】ウエーブレット変換のデコード装置の例を説明するための図である。

【図21】ウエーブレット変換のデコード垂直方向処理を説明するための図である。

【図22】ウエーブレット変換のデコード水平方向処理を説明するための図である。

【図23】サブバンドの大きさよりプリシンクトサイズが大きい場合を説明するための図である。

【図24】パケットデータ用ポインタの例を説明するための図である。

【図25】パケットポインタとパケットデータのメモリマッピングの例を説明するための図である。

【図26】サブバンドの大きさよりプリシンクトサイズが小さい場合を説明するための図である。

【図27】タグツリーの構成例を説明するための図である。

【図28】タグツリーのインクルージョンの検出例を説明するための図である。

#### 【符号の説明】

##### 1 画像データ

1 0 DWT部

1 1 量子化器

1 2 係数モデリング部

1 3 算術符号器

1 4 符号形成部

1 5 符号データ

2 0 1タイルの画像データ

3 7 パケットヘッダー

3 8 エントロピー符号

4 0 符号化処理回路

4 1、6 1 DCシフト・RCT変換回路

4 2、6 2 離散ウェーブレット変換回路(DWT回路)

4 3、6 3 算術符号化回路(MQ回路)

4 4 パケットヘッダ生成回路

4 5 符号列生成回路

4 6 メモリアクセスアビタ

4 7 SDRAMコントローラ

5 5 SDRAM

6 4 符号列解析回路

8 3～8 4 ラインバッファ

7 6～7 8 ハイパス処理回路

7 9～8 1 ローパス処理回路

8 5～9 0、1 0 8、2 0 8 クロック端子を有する  
フリップフロップ回路

7 1 入力端子

7 2～7 5 出力端子7 2～7 5

1 0 0～1 0 4、2 0 0～2 0 4 マルチブレクサ

1 0 5、1 0 6、2 0 5、2 0 6、2 2 4 加算器

1 0 7 反転回路

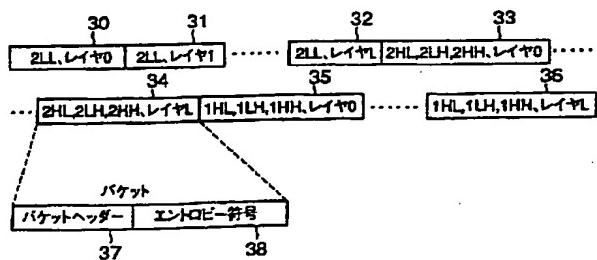
【図1】

JPEG2000の符号化の流れを説明するための図



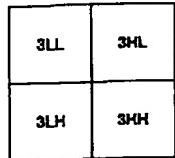
【図5】

ピットストリームを説明するための図



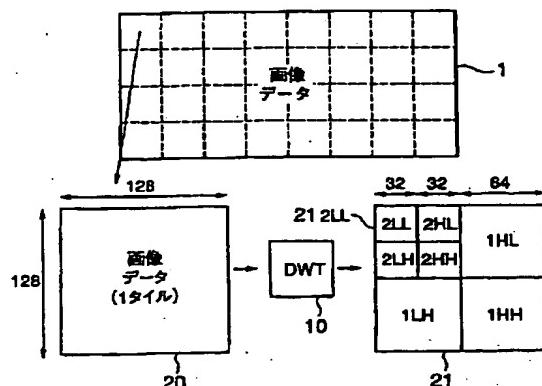
【図11】

3レベルのウェーブレット係数を説明するための図



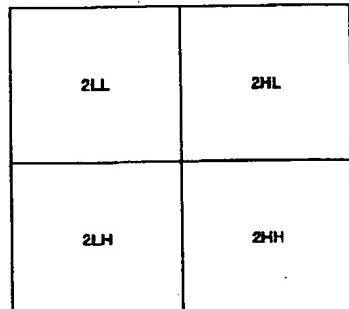
【図2】

画像データ1を128×128の大きさのタイルに分割し、レベル2の2次元離散ウェーブレット変換を行った場合の例を説明するための図



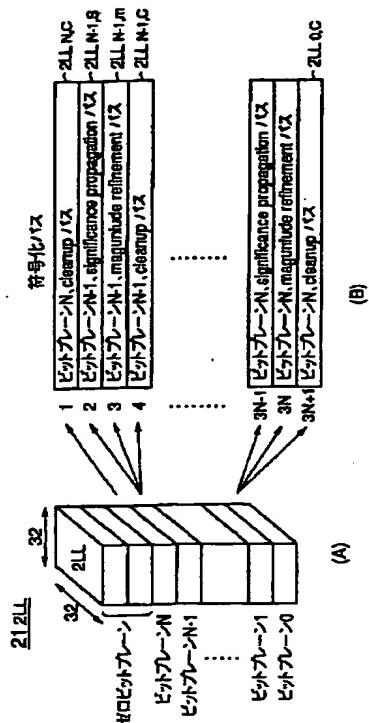
【図10】

2レベルのウェーブレット係数を説明するための図



【図3】

各サブバンド毎にエントロピー符号化した場合における  
サブバンド $2^{1L}$ とその算術符号化器の処理を説明するための図



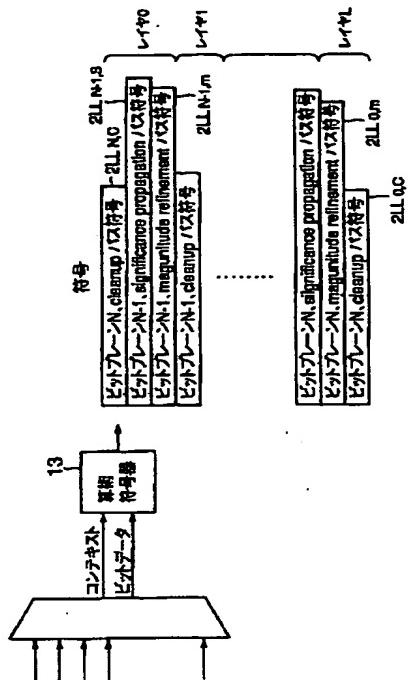
【図6】

3色（コンポーネント0、1、2）より成る画像をJPEG2000で  
符号化した場合のビットストリームの図

|                 |   |   |       |   |
|-----------------|---|---|-------|---|
| 30 <sub>1</sub> | コンポーネント0, 2 <sub>L</sub><br>レイヤ0                        | コンポーネント0, 2 <sub>L</sub><br>レイヤ1                        | ..... | コンポーネント0, 2 <sub>L</sub><br>レイヤL                        |
| 30 <sub>2</sub> | コンポーネント1, 2 <sub>L</sub><br>レイヤ0                        | コンポーネント1, 2 <sub>L</sub><br>レイヤ1                        | ..... | コンポーネント1, 2 <sub>L</sub><br>レイヤL                        |
| 30 <sub>3</sub> | コンポーネント2, 2 <sub>L</sub><br>レイヤ0                        | コンポーネント2, 2 <sub>L</sub><br>レイヤ1                        | ..... | コンポーネント2, 2 <sub>L</sub><br>レイヤL                        |
|                 | コンポーネント0, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤ0 | コンポーネント0, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤ1 | ..... | コンポーネント0, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤL |
|                 | コンポーネント1, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤ0 | コンポーネント1, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤ1 | ..... | コンポーネント1, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤL |
|                 | コンポーネント2, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤ0 | コンポーネント2, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤ1 | ..... | コンポーネント2, 2H <sub>L</sub><br>2L, 2H <sub>L</sub> , レイヤL |
| 36 <sub>1</sub> | コンポーネント0, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤ0 | コンポーネント0, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤ1 | ..... | コンポーネント0, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤL |
| 36 <sub>2</sub> | コンポーネント1, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤ0 | コンポーネント1, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤ1 | ..... | コンポーネント1, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤL |
| 36 <sub>3</sub> | コンポーネント2, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤ0 | コンポーネント2, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤ1 | ..... | コンポーネント2, 1H <sub>L</sub><br>1L, 1H <sub>L</sub> , レイヤL |

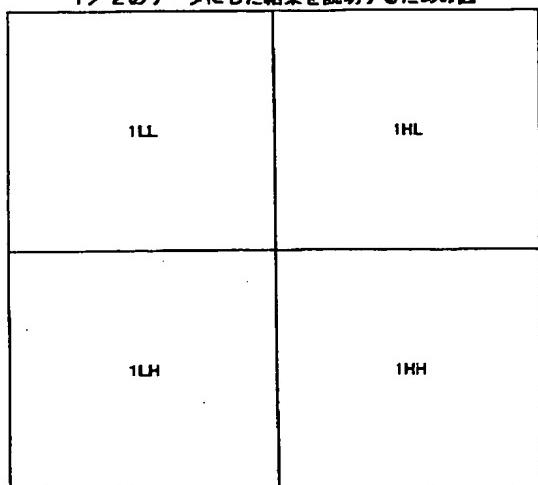
【図4】

SN+1個のバス毎に算術符号化器で生成された符号をレイヤ0から  
レイヤLまでのL+1個のレイヤにまとめている様子を示す図



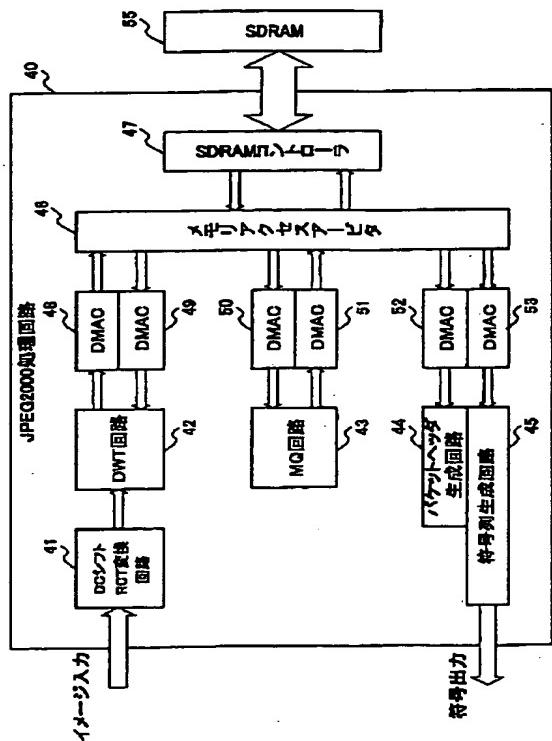
【図9】

垂直成分と水平成分に対して  
それぞれハイパスフィルタとローパスフィルタをかけて  
1/2のデータにした結果を説明するための図



【図7】

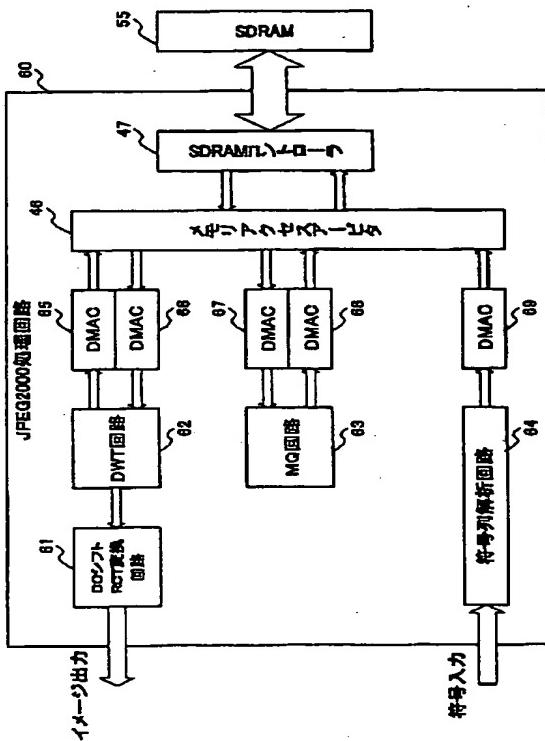
JPEG2000の符号化処理回路を説明するための図



【図12】

【図8】

JPEG2000の復号処理回路を説明するための図



【図13】

4レベルのウェーブレット係数を説明するための図



【図24】

5レベルのウェーブレット係数を説明するための図

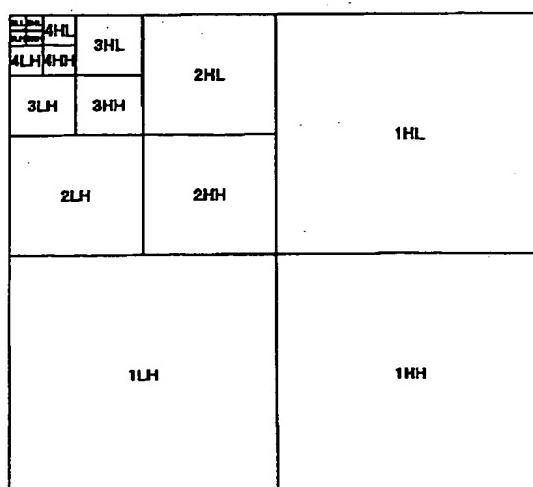


【図14】

パケットデータ用ポインタの例を説明するための図

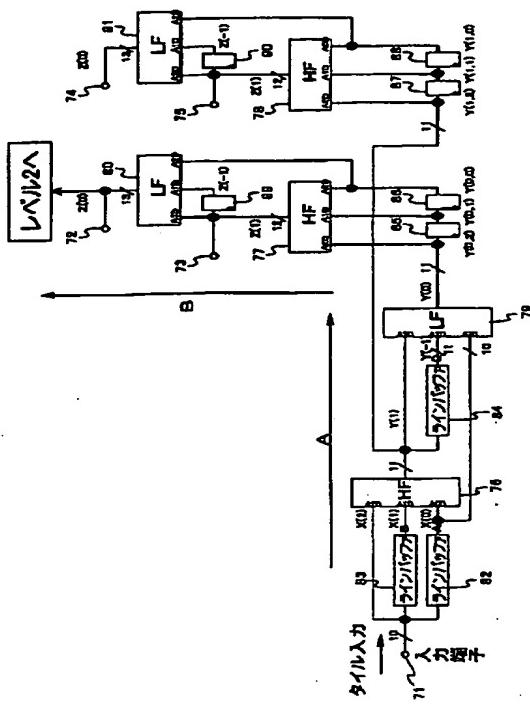
|       |             |
|-------|-------------|
| +0x00 | スタートアドレス    |
| +0x01 | スタートアドレス1   |
| +0x02 | スタートアドレス2   |
| +0x03 | スタートアドレス3   |
| +0x04 | データ長(バイト数)  |
| +0x05 | データ長(バイト数)  |
| +0x06 | ZEROビットブレーン |
| +0x07 | コーディングバース   |

最終的なウェーブレット変換のエンコードの結果を示す図



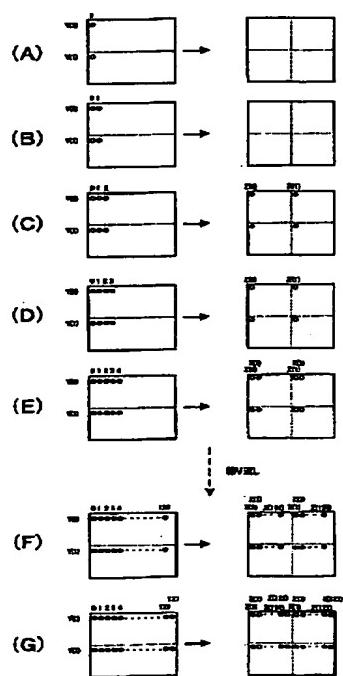
【図15】

ウェーブレット変換のエンコード装置を説明するための図



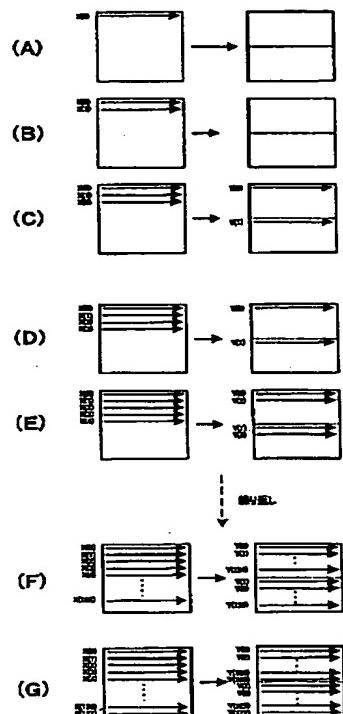
【図17】

ウェーブレット変換のエンコード水平方向処理を説明するための図



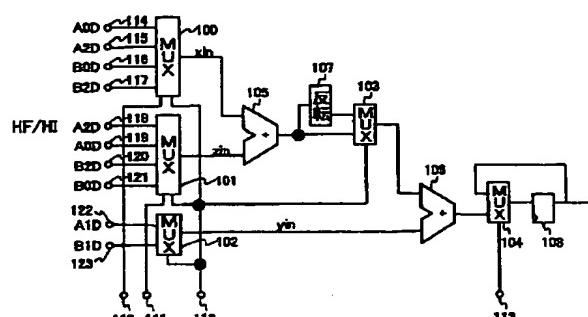
【図16】

ウェーブレット変換のエンコード垂直方向処理を説明するための図



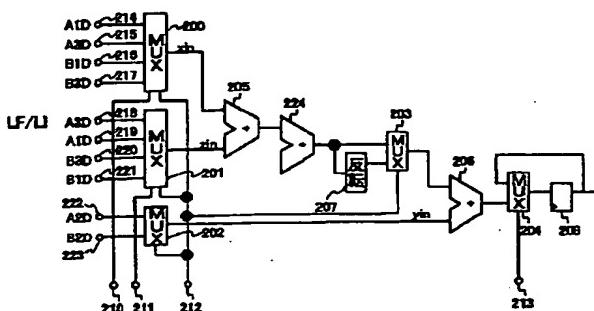
【図18】

ハイパス処理回路の例を説明するための図



【図19】

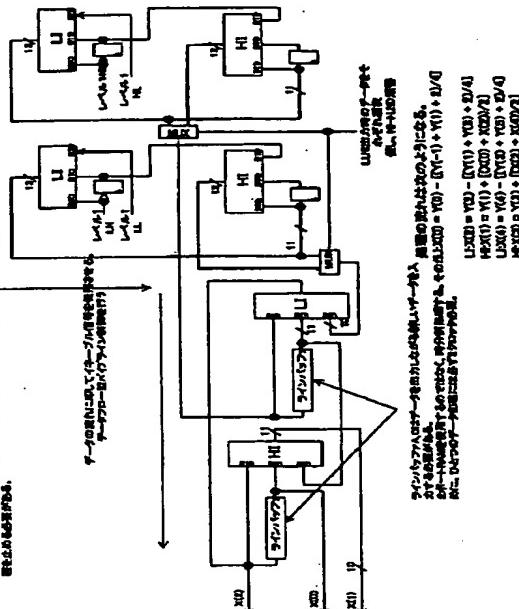
ローパス処理回路の例を説明するための図



【図21】

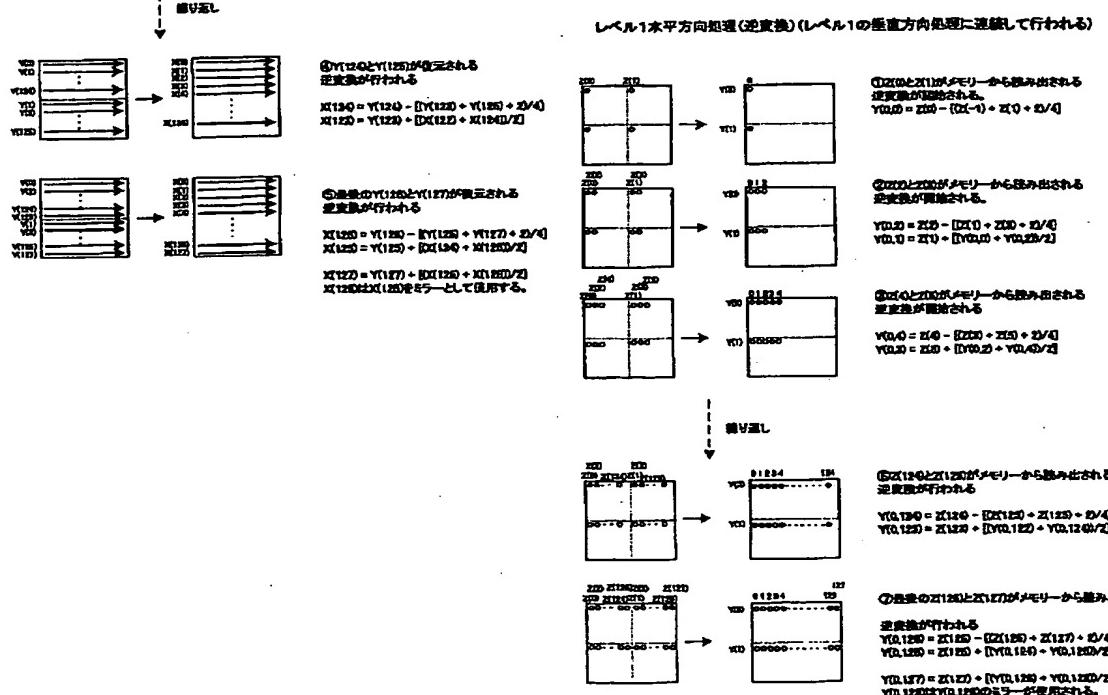
【図20】

ウェーブレット変換のデコード装置の例を説明するための図



【図22】

ウェーブレット変換のデコード水平方向処理を説明するための図



この処理をY[1,0]に及ぼしても行う。これらをモザイク処理。

【図23】

サブバンドの大きさよりプリシンクトサイズが  
大きい場合を説明するための図

| 0 1 | 4   | 7   | 8   | 18  | 20  | 21  | 22  | 67  | 68  | 69  | 70  | 71  | 72  | 73  | 74  |    |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 2 3 | 5   | 6   | 9   | 10  | 23  | 24  | 25  | 28  | 75  | 76  | 77  | 78  | 79  | 80  | 81  | 82 |
| 11  | 12  | 15  | 16  | 27  | 28  | 29  | 30  | 83  | 84  | 85  | 86  | 87  | 88  | 89  | 90  |    |
| 10  | 13  | 17  | 18  | 31  | 32  | 33  | 34  | 91  | 92  | 93  | 94  | 95  | 96  | 97  | 98  |    |
| 12  | 14  | 18  | 19  | 31  | 32  | 33  | 34  | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 |    |
| 25  | 38  | 37  | 58  | 51  | 52  | 53  | 54  | 58  | 103 | 104 | 105 | 106 | 107 | 108 | 109 |    |
| 39  | 40  | 41  | 42  | 55  | 56  | 57  | 58  | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 |    |
| 43  | 44  | 45  | 46  | 59  | 60  | 61  | 62  | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 |    |
| 47  | 48  | 49  | 50  | 63  | 64  | 65  | 66  | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 |    |
| 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 |    |
| 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 |    |
| 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 |    |
| 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 219 | 220 | 221 | 222 | 223 | 224 | 225 | 226 |    |
| 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 |    |
| 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 |    |
| 178 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 |    |
| 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 251 | 252 | 253 | 254 | 255 | 256 | 257 | 258 |    |

パケット1のデータ:100  
 パケット2のデータ:101-1H-19E2-133  
 パケット3のデータ:2H-1H-19E4-55  
 パケット4のデータ:2H-1H-19E2-1031-1415-15  
 パケット5のデータ:2H-1H-19E2-1031-0415-0515-06  
 パケット6のデータ:2H-1H-19E1-57-120,121-194,195-258

【図26】

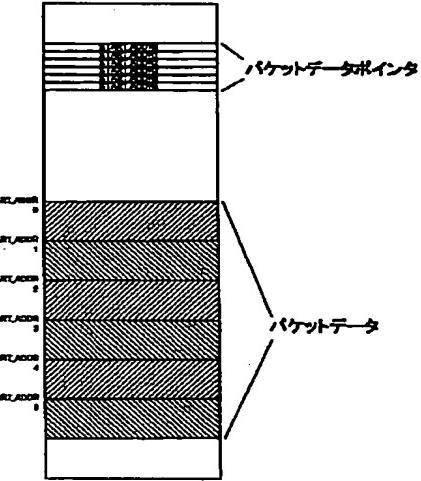
サブバンドの大きさよりプリシンクトサイズが  
小さい場合を説明するための図

| 0 1 | 4   | 7   | 8   | 18  | 20  | 21  | 22  | 67  | 68  | 69  | 70  | 71  | 72  | 73  | 74  |    |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 2 3 | 5   | 6   | 9   | 10  | 23  | 24  | 25  | 28  | 75  | 76  | 77  | 78  | 79  | 80  | 81  | 82 |
| 11  | 12  | 15  | 16  | 27  | 28  | 29  | 30  | 83  | 84  | 85  | 86  | 87  | 88  | 89  | 90  |    |
| 13  | 14  | 17  | 18  | 31  | 32  | 33  | 34  | 91  | 92  | 93  | 94  | 95  | 96  | 97  | 98  |    |
| 25  | 38  | 37  | 38  | 51  | 52  | 53  | 54  | 89  | 100 | 101 | 102 | 103 | 104 | 105 | 106 |    |
| 39  | 40  | 41  | 42  | 55  | 56  | 57  | 58  | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 |    |
| 43  | 44  | 45  | 46  | 59  | 60  | 61  | 62  | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 |    |
| 47  | 48  | 49  | 50  | 63  | 64  | 65  | 66  | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 |    |
| 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 |    |
| 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 |    |
| 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 |    |
| 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 219 | 220 | 221 | 222 | 223 | 224 | 225 | 226 |    |
| 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 |    |
| 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 |    |
| 178 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 |    |
| 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 251 | 252 | 253 | 254 | 255 | 256 | 257 | 258 |    |

パケット1のデータ:100  
 パケット2のデータ:2H-1H-19E2-133  
 パケット3のデータ:2H-1H-19E4-55  
 パケット4のデータ:2H-1H-19E2-1031-1415-15  
 パケット5のデータ:2H-1H-19E2-1031-0415-0515-06  
 パケット6のデータ:2H-1H-19E1-57-120,121-194,195-258

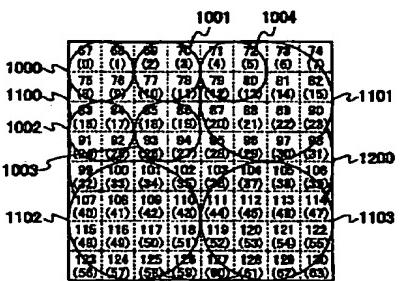
【図25】

パケットポインタとパケットデータのメモリマッピングの  
例を説明するための図



【図27】

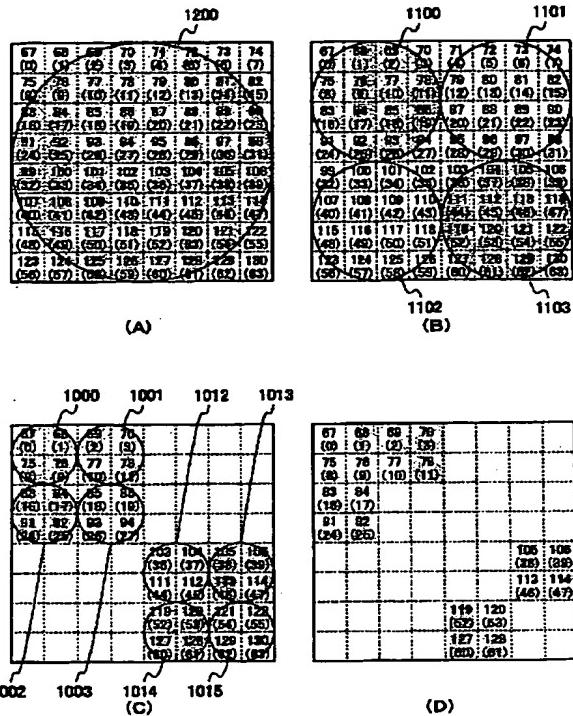
タグツリーの構成例を説明するための図



67,68,75,76がひとつのタグ(1000)になる。タグ1000,1001,1002,1003が集まって次の階層のタグ(1100)になる。1100,1101,1102,1103のタグが集まって次の次段のタグ(1200)になる。ひとつのタグから見て、タグを構成しているコードブロックまたはタグにすべてデータが含まれていない場合、そのタグにデータが含まれないことになる。タグを構成しているコードブロックまたはタグのどれかひとつにデータが含まれている場合、そのタグにデータが含まれていることになる。

【図28】

タグツリーのインクルージョンの検出例を説明するための図



フロントページの続き

Fターム(参考) 5B061 BB08 BC02  
 5C059 KK08 KK11 MA00 MA24 MC11  
 MC38 ME01 ME11 PP04 PP16  
 UA02 UA15 UA30 UA36 UA38  
 5C078 AA04 BA53 CA27 CA31 CA34  
 DA01 DA02